

RA-5557
10/607,885

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	TITUS ET AL.	Examiner:	Channavajjala, S.
Serial No.:	10/607,885	Group Art Unit:	2166
Filed:	June 27, 2003	Docket No.:	RA-5557 (USYS.069PA)
Title:	DYNAMIC FILE SIZE EXPANSION		

AMENDMENT AND RESPONSE

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

In response to the Office Action dated February 14, 2006, please consider the remarks that follow. Reconsideration and allowance of the application are respectfully requested.

Amendments to the Claims begin on page 2.

The Remarks begin on page 9.

AMENDMENTS TO THE CLAIMS

This listing of claims replaces all prior versions and listings of claims in the application.

Listing of Claims:

1. (Currently amended) A computer-implemented method for expanding usable space for an application data file, comprising:

maintaining in a control file first and second control structures and first and second pointers to the first and second control structures, respectively, for the data file, wherein the first structure includes a plurality of pointers that respectively reference ~~one or more~~ a plurality of bit maps that indicate available and allocated portions of usable space in the data file, and the second structure contains respective values that indicate ~~a quantity~~ quantities of available space in ~~[[a]]~~ portions of the data file;

limiting access to the first and second control structures to only a process that is expanding the data file while the process is expanding the data file;

allocating space for new versions of the first and second control structures in the control file;

copying contents of the first and second control structures to space for the new versions of the first and second control structures; and

updating the first and second pointers to reference the new versions of the first and second control structures.

2. (Original) The method of claim 1, further comprising:

maintaining in-memory copies of the first and second pointers while the application data file is available for access; and

updating the in-memory copies of the first and second pointers to reference the new versions of the first and second control structures.

3. (Original) The method of claim 1, wherein in a multi-host data processing arrangement for sharing the application data file, the method further comprises, after a first host completes expanding the data file, transmitting a message from the first host to each other host, wherein the message indicates that the file has been expanded.

4. (Original) The method of claim 1, further comprising:

determining whether the first structure is large enough to accommodate expansion of the data file by a requested amount; and

performing the steps of allocating, copying, and updating only if the first structure is not large enough to accommodate expansion of the data file by the requested amount.

5. (Original) The method of claim 1, wherein a plurality of application programs share the data file and are hosted on a plurality of host data processing systems (hosts), the method further comprising conditioning expansion of a file shared by the application program on whether each host is configured to detect expansion of the data file by another host.

6. (Original) The method of claim 5, further comprising, before expanding the data file by a first host, sending to each other host a message that queries whether the other host is configured to detect expansion of the data file by another host.

7. (Currently amended) A computer-implemented method for expanding usable space for an application data file, comprising:

maintaining an in-memory copy of one or more selected control structures from a control file while the application data file is available for access, wherein the application file is logically divided into a plurality of equal-size cells, and each cell provides storage for one or more records of data;

maintaining in the control file a first structure that contains pointers to second and third structures in the control file, wherein the second structure includes a plurality of pointers that respectively reference bit maps of the cells of the application file, each bit map indicating available and allocated records in the cell, and the third structure contains respective values that indicate **[[a quantity]]** quantities of available space in **[[a]]** the cells, and;

locking the first and third structures within the control file;

allocating in the control file space for a fourth structure and space for a fifth structure, wherein the space allocated for the fourth structure is greater than space occupied by the second structure, and the space allocated for the fifth structure is greater than space occupied by the third structure;

copying data from the second structure to the fourth structure and data from the third structure to the fifth structure;

updating in the first structure respective pointers to the second and third structures to reference the fourth and fifth structures, respectively; and

unlocking the first structure and the third structure after the pointers have been updated.

8. (Original) The method of claim 7, further comprising in response to a request that references an entry in the in-memory version of the third structure:

locking the third structure;

comparing a control file version of the pointer to the control file version of third structure, to an in-memory version of the pointer to the control file version of the third structure;

if the control file version and in-memory version of the pointer are not equal, then updating the in-memory versions of the pointers to the control file versions of the second and third structures, with the control file versions of the pointers to the control file versions of the second and third structures;

copying the contents of the third structure in the control file to the in-memory version of the third structure; and

unlocking the third structure.

9. (Original) The method of claim 7, wherein in a multi-host data processing arrangement for sharing the application data file, the method further comprises, after a first host completes expanding the data file, transmitting a message from the first host to each other host, wherein the message indicates that the file has been expanded.

10. (Original) The method of claim 9, wherein each host receiving a message that indicates that a file has been expanded, performs the steps comprising:

- locking the in-memory version of the third structure;
- updating the in-memory versions of the pointers to the control file versions of the second and third structures, with the control file versions of the pointers to the control file versions of the second and third structures;
- copying the contents of the third structure in the control file to the in-memory version of the third structure; and
- unlocking the third structure.

11. (Original) The method of claim 7, further comprising:

- determining whether the second structure is large enough to accommodate expansion of the data file by a requested amount; and
- performing the steps of allocating, copying, and updating only if the second structure is not large enough to accommodate expansion of the data file by the requested amount.

12. (Original) The method of claim 7, wherein a plurality of application programs share the data file and are hosted on a plurality of host data processing systems (hosts), the method further comprising conditioning expansion of a file shared by the application program on whether each host is configured to detect expansion of the data file by another host.

13. (Original) The method of claim 12, further comprising, before expanding the data file by a first host, sending to each other host a message that queries whether the other host is configured to detect expansion of the data file by another host.

14. (Currently amended) An apparatus for expanding usable space for an application data file, comprising:

- means for maintaining an in-memory copy of one or more selected control structures from a control file while the application data file is available for access,

wherein the application file is logically divided into a plurality of equal-size cells, and each cell provides storage for one or more records of data;

means for maintaining in the control file a first structure that contains pointers to second and third structures in the control file, wherein the second structure includes a plurality of pointers that respectively reference bit maps of the cells of the application file, each bit map indicating available and allocated records in the cell, and the third structure contains respective values that indicate **[[a quantity]] quantities** of available space in **[[a]] the cells**, and;

means for locking the first and third structures within the control file;

means for allocating in the control file space for a fourth structure and space for a fifth structure, wherein the space allocated for the fourth structure is greater than space occupied by the second structure, and the space allocated for the fifth structure is greater than space occupied by the third structure;

means for copying data from the second structure to the fourth structure and data from the third structure to the fifth structure;

means for updating in the first structure respective pointers to the second and third structures to reference the fourth and fifth structures, respectively; and

means for unlocking the first structure and the third structure after the pointers have been updated.

15. (Currently amended) An apparatus for expanding usable space for an application data file, comprising:

means for maintaining in a control file first and second control structures and first and second pointers to the first and second control structures, respectively, for the data file, wherein the first structure includes a plurality of pointers that respectively reference **[[one or more]] a plurality of** bit maps that indicate available and allocated portions of usable space in the data file, and the second structure contains respective values that indicate **[[a quantity]] quantities** of available space in **[[a]] portions** of the data file;

means for limiting access to the first and second control structures to only a process that is expanding the data file while the process is expanding the data file;

means for allocating space for new versions of the first and second control structures in the control file;

means for copying contents of the first and second control structures to space for the new versions of the first and second control structures; and

means for updating the first and second pointers to reference the new versions of the first and second control structures.

16. (Currently amended) An article of manufacture for expanding usable space for an application data file, comprising:

a computer-readable medium configured with instructions that cause a processor-based system to perform the steps of,

maintaining in a control file first and second control structures and first and second pointers to the first and second control structures, respectively, for the data file, wherein the first structure includes a plurality of pointers that respectively reference [[one or more]] a plurality of bit maps that indicate available and allocated portions of usable space in the data file, and the second structure contains respective values that indicate [[a quantity]] quantities of available space in [[a]] portions of the data file;

limiting access to the first and second control structures to only a process that is expanding the data file while the process is expanding the data file;

allocating space for new versions of the first and second control structures in the control file;

copying contents of the first and second control structures to space for the new versions of the first and second control structures; and

updating the first and second pointers to reference the new versions of the first and second control structures.

17. (Original) The article of manufacture of claim 16, wherein the computer-readable medium is further configured with instructions that cause a processor-based system to perform the steps of:

maintaining in-memory copies of the first and second pointers while the application data file is available for access; and

updating the in-memory copies of the first and second pointers to reference the new versions of the first and second control structures.

18. (Original) The article of manufacture of claim 16, wherein in a multi-host data processing arrangement for sharing the application data file, the computer-readable medium is further configured with instructions that cause a processor-based system to perform the step of transmitting a message from the first host to each other host after a first host completes expanding the data file, wherein the message indicates that the file has been expanded.

19. (Original) The article of manufacture of claim 16, wherein the computer-readable medium is further configured with instructions that cause a processor-based system to perform the steps of:

determining whether the first structure is large enough to accommodate expansion of the data file by a requested amount; and

performing the steps of allocating, copying, and updating only if the first structure is not large enough to accommodate expansion of the data file by the requested amount.

20. (Original) The article of manufacture of claim 16, wherein a plurality of application programs share the data file and are hosted on a plurality of host data processing systems (hosts), and the computer-readable medium is further configured with instructions that cause a processor-based system to perform the step of conditioning expansion of a file shared by the application program on whether each host is configured to detect expansion of the data file by another host.

21. (Original) The article of manufacture of claim 20, wherein the computer-readable medium is further configured with instructions that cause a processor-based system to perform the step of, before expanding the data file by a first host, sending to each other host a message that queries whether the other host is configured to detect expansion of the data file by another host.

REMARKS

Claims 1, 7, 14, 15, and 16 are amended to clarify the invention and not for purposes of patentability. Claims 1-21 remain for consideration, and reconsideration and allowance of the application are respectfully requested.

The Office Action fails to establish that claims 16-21 are unpatentable under 35 USC §101 as claiming non-statutory subject matter. The rejection is respectfully traversed because the Office Action does not establish a *prima facie* case that the invention as a whole is directed solely to an abstract idea or to manipulation of abstract ideas or does not produce a useful result.

The Office Action generally states that the recitation of “a computer readable medium” does not constitute a concrete or tangible result, and that amending the claims to read “a computer-readable storage medium” would remedy the alleged deficiency. It is respectfully submitted, however, that those skilled in the art would recognize that the added term “storage” is not needed to provide a tangible result. For example, the definition of “medium” at the Merriam-Webster Online Dictionary (<http://www.m-w.com/dictionary/medium>) provides “(4) : something (as a magnetic disk) on which information may be stored.” Thus, those skilled in the art will recognize that the added term “storage” is unnecessary, and perhaps even redundant. The rejection should be withdrawn.

The Office Action fails to establish that claims 1-6 and 14-21 are unpatentable under 35 U.S.C. 103(a) over “Loaiza” (“U.S. Patent No. 6,549,901 to Loaiza et al.”) in view of “Weisz” (U.S. Patent No. 6,128,621 to Weisz). The rejection is respectfully traversed because the Office Action fails to show that all the limitations are suggested by the references and fails to provide a proper motivation for modifying the teachings of Loaiza with teachings of Weisz.

The cited teachings of the Loaiza-Weisz combination neither teach nor suggest the limitations of the first structure including a plurality of pointers that respectively reference a plurality of bit maps that indicate available and allocated portions of usable space in the data file, and the second structure contains respective values that indicate

quantities of available space in portions of the data file.

It is not apparent what elements of Weisz are construed to correspond to the claimed bit maps. The term "bit map" does not even appear in the text of Weisz. Therefore Weisz neither teaches nor suggests bit maps that indicate available and allocated portions of usable space in the data file, or even bit maps.

It is not apparent what elements of Loaiza are construed to correspond to values that indicate quantities of available space in portions of the data file. Not only is there no apparent reference to a quantity of space relative to a data file in the cited teachings, but there is no apparent reference to any tracking of available space in a data file. The bit map limitations describe space in the data file as being available or allocated. Portions of the data file that are not allocated are available (p. 5, l. 3-9). Neither Loaiza nor Weisz in any apparent manner suggests these limitations.

If the rejection is maintained, clarification is respectfully requested. Specifically, recitation of those specific elements of Loaiza and Weisz that are construed as corresponding to the claim limitations is requested.

The cited teachings of Loaiza neither teach nor suggest the limitations of maintaining in a control file first and second control structures and first and second pointers to the first and second control structures.

Loaiza's table space 130 includes multiple data files. There is no apparent suggestion or indication of a control file. Furthermore, the limitations clearly set forth that the control file includes first and second pointers to the first and second control structures in the control file. There is no mention in the cited teachings of pointers in the control file which point to control structures in the control file. Loaiza's cited teachings reference disk pointers, not pointers that reference bit maps.

Independent claims 1 and 16 include the limitations discussed above, and claims 2-6, 14-15, and 17-21 include further limitations that further detail the limitations of the base claims. Therefore, the limitations of claims 1-6 and 14-21 are not shown to be suggested by the Loaiza-Weisz combination for at least the reasons set forth above. Furthermore, the alleged motivation for combining Weisz with Loaiza is conclusory and improper. The rejection of claims 1-6 and 14-21 over the Loaiza-Weisz combination should be withdrawn.

The Office Action fails to establish that claims 7-14 are unpatentable under 35 U.S.C. 103(a) over “Anfindsen” (“U.S. Patent No. 6,751,617 to Anfindsen) in view of Loaiza. The rejection is respectfully traversed because the Office Action fails to show that all the limitations are suggested by the references and fails to provide a proper motivation for modifying the teachings of Anfindsen with teachings of Loaiza.

The cited teachings of the Anfindsen-Loaiza combination neither teach nor suggest the limitations of a control file including a first structure that contains pointers to second and third structures in the control file, wherein the second structure includes a plurality of pointers that respectively reference bit maps of the cells of the application file, each bit map indicating available and allocated records in the cell, and the third structure containing respective values that indicate quantities of available space in the cells (claims 7 and 14).

It is not apparent what elements of Anfindsen are construed to correspond to the claimed bit maps. The term “bitmap” is taught by Anfindsen for use in indicating read parameters of a lock control block and for use as representing write parameters in a lock control block (col. 5, l. 38-49). The cited teachings contain no apparent and relevant suggestion of bit maps used to indicate available and allocated records in the cells. Therefore, the Office Action fails to show that Anfindsen teaches or suggests bit maps that indicate available and allocated portions of usable space in the data file.

Similarly, the cited sections of Anfindsen contain no elements that appear to be relevant to the claimed third structure with the values that indicate quantities of available space in the cells.

The limitations also clearly indicate that the control file includes pointers to the second and third structures in the control file. Since the Office Action does not clearly indicate which elements of Anfindsen are construed to correspond to the claimed bitmaps, and the teachings of Anfindsen do not appear to be relevant, the limitations of the pointers to the second and third structures are not shown to be suggested.

If the rejection is maintained, clarification is respectfully requested. Specifically, recitation of those specific elements of Anfindsen and Loaiza that are construed as corresponding to the claim limitations is requested.

Independent claims 7 and 14 include the limitations discussed above, and claims 8-13 include further limitations that further detail the limitations of the base claims. Therefore, the limitations of claims 7-14 are not shown to be suggested by the Anfindsen-Loaiza combination for at least the reasons set forth above. Furthermore, the alleged motivation for combining Loaiza with Anfindsen is conclusory and improper. The rejection of claims 7-14 Anfindsen-Loaiza combination should be withdrawn.

Withdrawal of the rejections and reconsideration of the claims are respectfully requested in view of the remarks set forth above. No extension of time is believed to be necessary for consideration of this response. However, if an extension of time is required, please consider this a petition for a sufficient number of months for consideration of this response. If there are any additional fees in connection with this response, please charge Deposit Account No. 50-0996 (USYS.069PA).

Respectfully submitted,

CRAWFORD MAUNU PLLC
1270 Northland Drive, Suite 390
Saint Paul, MN 55120
(651) 686-6633

By: _____
Name: LeRoy D. Maunu
Reg. No.: 35,274